



# ADVANCED PROGRAMMING INFORMATION

**Novation**



TM

18644 Oxnard Street, Tarzana, CA 91356 • (213) 996-5060

800155/882



## INTRODUCTION TO API

This manual serves as a guide to the advanced programmer. It describes RAM memory locations and their use as applies to the Apple-Cat II. Also included are some routines showing use of these memory locations, or registers as they're called, to control the modem. With this information, the advanced programmer can write or adapt machine language or BASIC routines to accomplish the individual tasks that he or she may desire.

The manual is organized by the registers, in ascending order, followed by the routines using them, collectively called 'CATPACK.' Each of the registers is broken down into its bit components and the function of each is described.

There are two indexes included in this appendix—one ordered numerically by register, the other ordered alphabetically by bit component description. Any register or bit component can be located easily. For example, if the register name or address is known, information in the first index will indicate where it can be found; if a particular bit component is needed, the second index will indicate its whereabouts.

'CATPACK' is a package of routines that shows how to control the Apple-Cat II through machine language. These routines show the use of the registers described in the previous section. One may want to modify them to meet specific applications requirements.



## REGISTER INDEX

### Base Address\*

| Hex                 | Decimal | Read   | Write  | Page  |
|---------------------|---------|--------|--------|-------|
| <b>APPLE-CAT II</b> |         |        |        |       |
| \$C080              | -16256  | SWBYT  | DACBYT | 4,5   |
| \$C081              | -16255  | TONBYT | SQUBYT | 5,7   |
| \$C082              | -16254  | —      | SHBYT  | 8     |
| \$C083              | -16253  | —      | BSRBYT | 9     |
| \$C089              | -16247  | —      | RECBYT | 10    |
| \$C08A              | -16246  | —      | CON    | 11    |
| \$C08B              | -16245  | INDAT  | SPDBYT | 12    |
| \$C08C              | -16244  | —      | COM    | 13    |
| \$C08D              | -16243  | ACBYT  | XMTBYT | 14,15 |
| \$C08E              | -16242  | —      | OUTDAT | 15    |
| \$C08F              | -16241  | STATUS | DEVBYT | 16,17 |
| <b>212 CARD</b>     |         |        |        |       |
| \$C080              | -16256  | —      | 212BYT | 18    |

\*Add \$N0 to hex base address or 16\*N to decimal base address (where N = slot of Apple-Cat II or 212 card) to obtain slot-adjusted address.

## BIT COMPONENT INDEX

| Bit Component         | Register | Page |
|-----------------------|----------|------|
| Audio detect          | SWBYT    | 4    |
| BSR 60 Hz input       | SWBYT    | 4    |
| BSR 120 KHz output    | BSRBYT   | 9    |
| Carrier detect        | SWBYT    | 4    |
| Carrier detect timing | RECBYT   | 10   |
| Clear DV              | SHBYT    | 8    |
| Clear to send         | ACBYT    | 14   |
| DAC bits              | DACBYT   | 5    |
| DV from TouchTone     | SWBYT    | 4    |
| Firmware switches     | SWBYT    | 4    |
| From UART txd to      | DEVBYT   | 17   |
| IRQ ring enable       | XMTBYT   | 15   |
| IRQ 29.12 Hz enable   | BSRBYT   | 9    |
| IRQ 29.12 Hz status   | TONBYT   | 5    |
| Microphone squelch    | SQUBYT   | 7    |



Continued

### **Bit Component**

|                                       | <b>Register</b> | <b>Page</b> |
|---------------------------------------|-----------------|-------------|
| Mode control bits                     | RECBYT          | 10          |
| Receive baud rate                     | SPDBYT          | 12          |
| Receive data from UART                | INDAT           | 12          |
| Ring detect                           | ACBYT           | 14          |
| Speaker squelch                       | SQUBYT          | 7           |
| Switch hook                           | SHBYT           | 8           |
| Tape recorder motor control           | SQUBYT          | 7           |
| TouchTone bits                        | TONBYT          | 5           |
| Transmit baud rate                    | SPDBYT          | 12          |
| Transmit data to UART                 | OUTDAT          | 15          |
| Transmitter mode                      | XMTBYT          | 15          |
| UART command—receiver                 | COM             | 13          |
| UART command—transmitter              | COM             | 11          |
| UART control—parity enable            | CON             | 11          |
| UART control—parity                   | CON             | 11          |
| UART control—# stop bits              | CON             | 11          |
| UART control—# data bits              | CON             | 11          |
| UART rxd to                           | DEVBYT          | 17          |
| UART status—overrun error             | STATUS          | 16          |
| UART status—framing error             | STATUS          | 16          |
| UART status—parity error              | STATUS          | 16          |
| UART status—tx data<br>register empty | STATUS          | 16          |
| UART status—rx data<br>register full  | STATUS          | 16          |
| UART status—transmit IRQ              | STATUS          | 16          |
| UART status—receive IRQ               | STATUS          | 16          |
| UART status—ring IRQ                  | STATUS          | 16          |
| 212 analog loopback                   | 212BYT          | 18          |
| 212 carrier detect                    | ACBYT           | 14          |
| 212 character length                  | 212BYT          | 18          |
| 212 digital loopback                  | 212BYT          | 18          |
| 212 enable                            | SQUBYT          | 7           |
| 212 mode                              | SQUBYT          | 7           |
| 212 remote digital loopback           | 212BYT          | 18          |
| 212 reset option                      | 212BYT          | 18          |



### **SWBYT \$C080 -16256 READ**

|                                  |                            |                        |                      |          |          |          |          |
|----------------------------------|----------------------------|------------------------|----------------------|----------|----------|----------|----------|
| 7<br>DV<br>FROM<br>TOUCH<br>TONE | 6<br>BSR<br>60 HZ<br>INPUT | 5<br>CARRIER<br>DETECT | 4<br>AUDIO<br>DETECT | 3<br>SW4 | 2<br>SW3 | 1<br>SW2 | 0<br>SW1 |
|----------------------------------|----------------------------|------------------------|----------------------|----------|----------|----------|----------|

#### **Bit**

- 7 DV FROM TOUCHTONE
  - 1 valid TouchTone present
  - 0 no valid TouchTone present (CLRDV high)
- 6 BSR 60 HZ INPUT
  - continuous 1 no 60 Hz
  - alternating 60 Hz
- 5 CARRIER DETECT
  - 1 carrier present
  - 0 no carrier present
- 4 AUDIO DETECT\*
  - 1 no audio present
  - 0 audio present
- 3 FIRMWARE SWITCH 4
  - 1 open (off)
  - 0 closed (on)
- 2 FIRMWARE SWITCH 3
  - 1 open (off)
  - 0 closed (on)
- 1 FIRMWARE SWITCH 2
  - 1 open (off)
  - 0 closed (on)
- 0 FIRMWARE SWITCH 1
  - 1 open (off)
  - 0 closed (on)

\*Audio detect input changes with speech, dial tone, and busy signal. The input to the audio detector is the output of the receive filter.



**DACBYT**  
**\$C080**  
**-16256**  
**WRITE**

|                                      |   |   |   |   |   |   |                                       |
|--------------------------------------|---|---|---|---|---|---|---------------------------------------|
| 7<br>MOST<br>SIGNIFI-<br>CANT<br>BIT | 6 | 5 | 4 | 3 | 2 | 1 | 0<br>LEAST<br>SIGNIFI-<br>CANT<br>BIT |
|--------------------------------------|---|---|---|---|---|---|---------------------------------------|

**8 DAC bits**

**TONBYT**  
**\$C081**  
**-16255**  
**READ**

|                  |                  |                  |                                   |                                 |                                 |                                 |                                 |
|------------------|------------------|------------------|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 7<br>Not<br>Used | 6<br>Not<br>Used | 5<br>Not<br>Used | 4<br>IRO<br>29.12<br>HZ<br>STATUS | 3<br>TOUCH<br>TONE<br>BIT<br>D8 | 2<br>TOUCH<br>TONE<br>BIT<br>D4 | 1<br>TOUCH<br>TONE<br>BIT<br>D2 | 0<br>TOUCH<br>TONE<br>BIT<br>D1 |
|------------------|------------------|------------------|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|

**Bit**

- 7 Not used
- 6 Not used
- 5 Not used
- 4 IRQ 29.12 HZ STATUS
  - 1 IRQ has not occurred
  - 0 IRQ has occurred

**3-0 TOUCHTONE BITS**

| digit | D8 | D4 | D2 | D1 |
|-------|----|----|----|----|
| 1     | 0  | 0  | 0  | 1  |
| 2     | 0  | 0  | 1  | 0  |
| 3     | 0  | 0  | 1  | 1  |
| 4     | 0  | 1  | 0  | 0  |
| 5     | 0  | 1  | 0  | 1  |
| 6     | 0  | 1  | 1  | 0  |
| 7     | 0  | 1  | 1  | 1  |



| digit | D8 | D4 | D2 | D1 |
|-------|----|----|----|----|
| 8     | 1  | 0  | 0  | 0  |
| 9     | 1  | 0  | 0  | 1  |
| 0     | 1  | 0  | 1  | 0  |
| *     | 1  | 0  | 1  | 1  |
| #     | 1  | 1  | 0  | 0  |
| A     | 1  | 1  | 0  | 1  |
| B     | 1  | 1  | 1  | 0  |
| C     | 1  | 1  | 1  | 1  |
| D     | 0  | 0  | 0  | 0  |

**TIMING:** Within 40 ms of a valid received tone pair the data bits D8, D4, D2, and D1 will become valid. Seven microseconds after the data bits have become valid, DV will go high. The data bits will remain valid and DV will remain high as long as a valid tone pair is present at the receiver input. Within 40 ms of the removal of a valid tone pair from the input, the decoder will recognize a valid pause. DV goes low approximately 45 ms after the tone pair is removed. The data bits will be cleared (set to 0) 4.4 ms after DV is lowered. The DV strobe will be of at least the same duration as the incoming tone pair.

**HANDSHAKE MODE:** In this mode, the DV strobe is polled at least once every 40 ms to determine whether a new valid tone pair has been detected. If DV is high, indicating that a new valid tone pair has been detected, store the coded data present at the data bits of the receiver and clear DV by pulsing CLRDV high. With some systems operating in the handshake mode, it may be desirable to know when the valid tone has ended. Ordinarily this would be indicated by the falling edge of DV. However, in the handshake mode, DV is cleared by the monitoring system each time a new valid tone is detected and, therefore, cannot be used to determine when a valid tone pair has ended. The termination of a valid tone pair in this case may be observed by detecting the clearing of the data bits. Since, in hexadecimal format (the mode normally used with a handshake interface), the all zero state represents a commonly unused tone pair (D), the end of a valid tone pair may be detected by sensing the all zero state.

**NOTE:** DV is read as SWBYT bit 7. DV is cleared in SHBYT bit 0.



**SQUBYT**  
**\$C081**  
**-16255**  
**WRITE**

|                    |                  |                  |           |           |           |          |          |
|--------------------|------------------|------------------|-----------|-----------|-----------|----------|----------|
| 7<br>212<br>ENABLE | 6<br>Not<br>Used | 5<br>212<br>MODE | 4<br>TRMC | 3<br>TRMC | 2<br>TRMC | 1<br>SSQ | 0<br>MSQ |
|--------------------|------------------|------------------|-----------|-----------|-----------|----------|----------|

**Bit**

- 7 212 ENABLE\*
  - 1 212 disable
  - 0 212 enable
- 6 Not used
- 5 212 MODE
  - 1 212 originate
  - 0 212 answer
- 4-2 TAPE RECORDER MOTOR CONTROL
  - 111 motor on
  - 000 motor off
- 1 SPEAKER SQUELCH
  - 1 not squelched
  - 0 squelched
- 0 MICROPHONE SQUELCH
  - 1 squelched
  - 0 not squelched

\*212 ENABLE is set to 1 in 212 analog loopback test (see 212BYT).

NOTE: When changing ANS/ORIG mode, first write word with bit 5 in desired state and with bit 7 high; then wait 5-10 msec, and write word with same state for bit 5 but with bit 7 low. Bit 7 must be left enabled (low) whenever operating in 212.

**SHBYT**  
**\$C082**  
**-16254**  
**WRITE**

|                  |                  |                  |                  |                  |                  |         |            |
|------------------|------------------|------------------|------------------|------------------|------------------|---------|------------|
| 7<br>Not<br>Used | 6<br>Not<br>Used | 5<br>Not<br>Used | 4<br>Not<br>Used | 3<br>Not<br>Used | 2<br>Not<br>Used | 1<br>SH | 0<br>CLRDV |
|------------------|------------------|------------------|------------------|------------------|------------------|---------|------------|

**Bit**

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 Not used
- 2 Not used
- 1 SWITCH HOOK
  - 1 off hook (holding telephone line)
  - 0 on hook (not holding telephone line)
- 0 CLEAR DV (Reset TouchTone)
  - 1 clear DV to 0
  - 0 DV set



**BSRBYT**  
**\$C083**  
**-16253**  
**WRITE**

|               |               |               |               |               |                                |                               |               |
|---------------|---------------|---------------|---------------|---------------|--------------------------------|-------------------------------|---------------|
| 7<br>Not Used | 6<br>Not Used | 5<br>Not Used | 4<br>Not Used | 3<br>Not Used | 2<br>IRQ<br>29.12 HZ<br>ENABLE | 1<br>BSR<br>120 KHZ<br>OUTPUT | 0<br>Not Used |
|---------------|---------------|---------------|---------------|---------------|--------------------------------|-------------------------------|---------------|

**Bit**

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 Not used
- 2 IRQ 29.12 HZ ENABLE
  - 1 IRQ disabled/reset\*
  - 0 IRQ enabled
- 1 BSR 120 KHZ OUTPUT
  - 1 disabled
  - 0 enabled
- 0 Not used

\*To clear IRQ, Bit 2 must be disabled, then enabled immediately.



**RECBYT**  
**\$C089**  
**-16247**  
**WRITE**

|               |          |          |          |          |          |          |          |
|---------------|----------|----------|----------|----------|----------|----------|----------|
| 7<br>Not Used | 6<br>CDT | 5<br>MC6 | 4<br>MC5 | 3<br>MC4 | 2<br>MC3 | 1<br>MC2 | 0<br>MC1 |
|---------------|----------|----------|----------|----------|----------|----------|----------|

**Bit**

- 7 Not used
- 6 CARRIER DETECT TIMING\*
  - 1 normal carrier
  - 0 fast carrier

5-0 MODE CONTROL BITS

| mode                            | value |
|---------------------------------|-------|
| 103 ANS                         | \$64  |
| 103 ANS (mark carrier detect)   | \$60  |
| 103 ORIG                        | \$65  |
| 103 ORIG (mark carrier detect)  | \$61  |
| 202                             | \$5C  |
| 202 (mark carrier detect)       | \$58  |
| DEAF (TDD)                      | \$2D  |
| ANS V.21                        | \$66  |
| ANS V.21 (mark carrier detect)  | \$62  |
| ORIG V.21                       | \$67  |
| ORIG V.21 (mark carrier detect) | \$63  |
| 900 Hz detect                   | \$6F  |
| dial tone & busy signal †       | \$7E  |

\*Normal carrier

- 103, V.21: on delay—150ms
- off delay— 50ms
- 202, V.23: on delay— 38ms
- off delay— 13ms

Fast carrier (must be used in TDD mode)

- All modes: on delay—9ms
- off delay—9ms

†This mode does not detect dial tone or busy but simply puts the receive filter in the frequency range of dial tone, busy signals, and ringing. The audio detect bit (SWBYT bit 4) should be looked at to determine what is dial tone, busy, or speech.



**CON**  
**\$C08A**  
**-16246**  
**WRITE**

**UART CONTROL BYTE**

|               |         |         |         |          |          |          |          |
|---------------|---------|---------|---------|----------|----------|----------|----------|
| 7<br>Not Used | 6<br>PE | 5<br>P2 | 4<br>P1 | 3<br>SB2 | 2<br>SB1 | 1<br>DB2 | 0<br>DB1 |
|---------------|---------|---------|---------|----------|----------|----------|----------|

**Bit**

- 7 Not used
- 6 PARITY ENABLE
  - 1 enabled
  - 0 disabled

5–4 PARITY\*

- 11 mark
- 10 odd
- 01 space
- 00 even

3–2 NUMBER OF STOP BITS

- 11 —
- 10 2
- 01 1.5
- 00 1

1–0 NUMBER OF DATA BITS

- 11 8
- 10 6
- 01 7
- 00 5

\*The parity bit is additional to the number of data bits.



**INDAT**  
**\$C08B**  
**-16245**  
**READ**

**RECEIVE DATA FROM UART**

|                                      |   |   |   |   |   |   |                                       |
|--------------------------------------|---|---|---|---|---|---|---------------------------------------|
| 7<br>MOST<br>SIGNIFI-<br>CANT<br>BIT | 6 | 5 | 4 | 3 | 2 | 1 | 0<br>LEAST<br>SIGNIFI-<br>CANT<br>BIT |
|--------------------------------------|---|---|---|---|---|---|---------------------------------------|

**SPDBYT**  
**\$C08B**  
**-16245**  
**WRITE**

|           |           |           |           |           |           |           |           |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 7<br>TBR4 | 6<br>TBR3 | 5<br>TBR2 | 4<br>TBR1 | 3<br>RBR4 | 2<br>RBR3 | 1<br>RBR2 | 0<br>RBR1 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|

**Bit**

7–4 TRANSMIT BAUD RATE\*

3–0 RECEIVE BAUD RATE\*

| baud rate | bit time | actual baud rate | percent error | value |
|-----------|----------|------------------|---------------|-------|
| 1200      | .831 ms  | 1202.75          | .2            | \$0   |
| 600       | 1.66 ms  | 601.38           | .2            | \$1   |
| 300       | 3.33 ms  | 300.68           | .2            | \$2   |
| 150       | 6.65 ms  | 150.34           | .2            | \$3   |
| 110       | 9.15 ms  | 109.27           | .6            | \$5   |
| 75        | 13.3 ms  | 75.17            | .2            | \$6   |
| 50        | 20.0 ms  | 50.11            | .2            | \$7   |
| 45.5      | 22.0 ms  | 46.26            | 1.7           | \$4   |

NOTE: Transmit and receive baud rates are independent.



**COM  
\$C08C  
-16244  
WRITE**

**UART COMMAND REGISTER**

|                  |                  |                  |                  |         |         |         |         |
|------------------|------------------|------------------|------------------|---------|---------|---------|---------|
| 7<br>Not<br>Used | 6<br>Not<br>Used | 5<br>Not<br>Used | 4<br>Not<br>Used | 3<br>T2 | 2<br>T1 | 1<br>R2 | 0<br>R1 |
|------------------|------------------|------------------|------------------|---------|---------|---------|---------|

**Bit**

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used

**3–2 TRANSMITTER**

- 11 xmit off
- 10 xmit on, enable IRQ
- 01 xmit on, disable IRQ
- 00 xmit break (space)

**1–0 RECEIVER**

- 11 —
- 10 receive on, enable IRQ
- 01 receive on, disable IRQ
- 00 receiver off



**ACBYT  
\$C08D  
-16243  
READ**

|                  |                  |                  |                  |          |                               |                  |                     |
|------------------|------------------|------------------|------------------|----------|-------------------------------|------------------|---------------------|
| 7<br>Not<br>Used | 6<br>Not<br>Used | 5<br>Not<br>Used | 4<br>Not<br>Used | 3<br>CTS | 2<br>212<br>CARRIER<br>DETECT | 1<br>Not<br>Used | 0<br>RING<br>DETECT |
|------------------|------------------|------------------|------------------|----------|-------------------------------|------------------|---------------------|

**Bit**

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 CLEAR TO SEND  
printer handshake (busy)  
follows input
- 2 212 CARRIER DETECT
  - 1 carrier present
  - 0 no carrier present
- 1 Not used
- 0 RING DETECT\*
  - 1 ring present
  - 0 ring not present

\*If IRQ enabled on ring, IRQ occurs on leading edge of ring (when RING goes high).



**XMTBYT**  
**\$C08D**  
**-16243**  
**WRITE**

|   |          |   |          |   |          |   |                 |   |     |   |     |   |     |   |     |
|---|----------|---|----------|---|----------|---|-----------------|---|-----|---|-----|---|-----|---|-----|
| 7 | Not Used | 6 | Not Used | 5 | Not Used | 4 | IRQ RING ENABLE | 3 | XM4 | 2 | XM3 | 1 | XM2 | 0 | XM1 |
|---|----------|---|----------|---|----------|---|-----------------|---|-----|---|-----|---|-----|---|-----|

**Bit**

7 Not used

6 Not used

5 Not used

4 IRQ RING ENABLE\*

1 disabled/reset

0 enabled

## 3-0 TRANSMITTER MODE

| function               | value | space Hz | mark Hz     |
|------------------------|-------|----------|-------------|
| 103 ANS                | \$0   | 2025     | 2225        |
| 103 ORIG               | \$1   | 1070     | 1270        |
| CCITT ANS              | \$2   | 1850     | 1650        |
| CCITT ORIG             | \$3   | 1180     | 980         |
| DEAF (TDD)             | \$4   | 1800     | 1400        |
| 202 SOFT CAR           | \$5   | 900      |             |
| 202 U.S. or CCITT V.23 | \$6   | 2100     | 1300        |
| REV CHAN (VIEWDATA)    | \$7   | 450      | 390         |
| disabled               | >\$7  |          | off (quiet) |

\*To clear IRQ, write 1 then 0.

**OUTDAT**  
**\$C08E**  
**-16242**  
**WRITE**

**TRANSMIT DATA TO UART**

|   |                       |   |   |   |   |   |   |                          |
|---|-----------------------|---|---|---|---|---|---|--------------------------|
| 7 | MOST SIGNIFI-CANT BIT | 6 | 5 | 4 | 3 | 2 | 1 | 0 LEAST SIGNIFI-CANT BIT |
|---|-----------------------|---|---|---|---|---|---|--------------------------|



**STATUS**  
**\$C08F**  
**-16241**  
**READ**

**UART STATUS REGISTER**

|   |    |   |    |   |    |   |    |   |    |   |    |   |    |   |    |
|---|----|---|----|---|----|---|----|---|----|---|----|---|----|---|----|
| 7 | S8 | 6 | S7 | 5 | S6 | 4 | S5 | 3 | S4 | 2 | S3 | 1 | S2 | 0 | S1 |
|---|----|---|----|---|----|---|----|---|----|---|----|---|----|---|----|

**Bit**

7 OVERRUN ERROR

1 error

0 ok

6 FRAMING ERROR

1 error

0 ok

5 PARITY ERROR

1 error

0 ok

4 TX DATA REG EMPTY

1 empty

0 full

3 RX DATA REG FULL

1 full

0 empty

2 TRANSMIT IRQ

1 transmit IRQ

0 no transmit IRQ

1 RECEIVE IRQ

1 receive IRQ

0 no receive IRQ

0 RING IRQ

1 no IRQ

0 IRQ



**DEVBYT**  
**\$C08F**  
**-16241**  
**WRITE**

**INPUT/OUTPUT SELECTION REGISTER\***

|               |               |               |               |           |           |           |           |
|---------------|---------------|---------------|---------------|-----------|-----------|-----------|-----------|
| 7<br>Not Used | 6<br>Not Used | 5<br>Not Used | 4<br>Not Used | 3<br>TXD2 | 2<br>TXD1 | 1<br>RXD2 | 0<br>RXD1 |
|---------------|---------------|---------------|---------------|-----------|-----------|-----------|-----------|

**Bit**

7 Not used

6 Not used

5 Not used

4 Not used

3-2 FROM UART TXD TO

11 —

10 add-on 212

01 printer port

00 main modem

1-0 UART RXD TO

11 —

10 add-on 212

01 printer port

00 main modem

\*Register must be written to when entering 212 mode.



**212BYT**  
**\$C080**  
**-16256**  
**WRITE**

|               |               |         |         |          |          |          |          |
|---------------|---------------|---------|---------|----------|----------|----------|----------|
| 7<br>Not Used | 6<br>Not Used | 5<br>AL | 4<br>DL | 3<br>RDL | 2<br>CL2 | 1<br>CL1 | 0<br>RST |
|---------------|---------------|---------|---------|----------|----------|----------|----------|

**Bit**

7 Not used

6 Not used

5 ANALOG LOOPBACK

1 analog loopback

0 normal

4 DIGITAL LOOPBACK

1 digital loopback

0 normal

3 REMOTE DIGITAL LOOPBACK

1 remote digital loopback

0 normal

2-1 212 CHARACTER LENGTH\*

10 10 bits

01 9 bits

0 RESET

1 normal

0 reset option

\*Length includes one start and one stop bit.

NOTE: Procedure for writing

1) Write with bits as desired and bit 0 low.

2) Delay at least 500 ms.

3) Write with bit 0 high, all other bits unchanged.



LISA 2.5

## CATPACK

```

0800      2      TITL "CATPACK"
0800      3      ;A PACKAGE OF ROUTINES THAT WILL SERVE
0800      4      ;AS EXAMPLES OF HOW TO ACCESS THE
0800      5      ;APPLE-CAT MODEM THROUGH ASSEMBLY
0800      6      ;LANGUAGE. THESE ROUTINES ARE DESIGNED
0800      7      ;TO BE EXAMPLES, AND YOU MAY WANT TO
0800      8      ;MODIFY THEM TO MEET YOUR NEEDS.
0800      9      ;
0800     10      ;BY GREG SEITZ
0800     11      ;COPYRIGHT (C) 1981 NOVATION INC.
0800     12      ;
0800     13      ;REV 1.1 13-FEB-82
0800     14      ;ALL COMMERCIAL RIGHTS RESERVED
0800     15      ;
0800     16      ;NOTE: SOME OF THE STORAGE LOCATIONS
0800     17      ;USED IN THIS PROGRAM MAY CONFLICT WITH
0800     18      ;APPLE-SOFT OR INTEGER BASIC SO IF YOU
0800     19      ;WILL BE USING EITHER YOU MUST MAKE SURE
0800     20      ;THAT NO CONFLICTS ARISE...
0800     21      ;
0800     22      ;WARNING: MOST OF THESE ROUTINES WILL
0800     23      ;RETURN WITH THE REGISTERS WIPE'D OUT.
0800     24      ;SO IF YOU NEED TO SAVE THE VALUE
0800     25      ;OF A REGISTER YOU SHOULD DO SO BEFORE
0800     26      ;CALLING ANY OF THESE ROUTINES.
0800     27      ;
00E9    28      DLYL   EPZ $E9
00EA   29      DLYH   EPZ $EA
00E5   30      F1F    EPZ $E5
00E6   31      F1I    EPZ $E6
00E7   32      F2F    EPZ $E7
00FB   33      F2I    EPZ $E8
00B0   34      DACBYT EQU $C0B0
0095   35      CMDTAB EPZ $95
005D   36      CMD    EPZ $5D
0097   37      CMDBLOC EPZ $97
03FE   38      JRD    EOU $3FE
00E1   39      FRAC1  EPZ $E1
00E0   40      INI1   EPZ $E0
00E4   41      FRAC2  EPZ $E4
00E3   42      INT2  EPZ $E3
0004   43      C3    EPZ $04
0011   44      COUNT EPZ $11
001C   45      TEMPX EPZ $1C
001D   46      TEMPY EPZ $1D
001F   47      CARSTS EPZ $1F
00F6   48      ZCTMP  EPZ $F6
0002   49      CROSS  EPZ $02
FC58   50      HOME   EOU $FC58
0033   51      PROMPT EPZ $33
FD6A   52      GETLN  EOU $FD6A
0010   53      C7    EPZ $10
0200   54      KEYBUF EOU $200
0350   55      DIGBUF EOU $250
FCA8   56      WAIT   EOU #FCAB
030C   57      DIALSW EOU $30C
02F1   58      SLUT   EOU $2F1
0010   59      KEYCLR EOU $C010
001B   60      XMTRAM EPZ $1B

```

LISA 2.5

## CATPACK

```

C000      61      KEYDAT  EOU $C000
C00C      62      COM     EOU $C08C
C00D      63      XMTBYT EOU $C08D
C08A      64      CON     EOU $C08A
C083      65      BSRTBYT EOU $C083
C082      66      SHBYT   EOU $C082
C081      67      TONBYT  EOU $C081
C081      68      SOUBYT  EOU $C081
C080      69      SWBYT   EOU $C080
C089      70      RECBYT  EOU $C089
C08B      71      INDAT   EOU $C08B
C08E      72      OUTDAT  EOU $C08E
C08B      73      SPDIBYT EOU $C08B
C08F      74      STATUS   EOU $C08F
C08D      75      ACRYT   EOU $C08D
C08F      76      DEVBYT  EOU $C08F
F3ED      77      COU    EOU $F3ED
FDF0      78      COUT1   EOU $FDF0
0800      79      ;
0800      80      ;
0800      81      ; THE FOLLOWING ROUTINE IS A TYPICAL
0800      82      ;EXAMPLE OF WHAT YOU WOULD DO UPON
0800      83      ;ENTRY TO A MODEM PROGRAM. FIRST WE
0800      84      ;GET THE SLOT THEN WE SAVE IT AND
0800      85      ;JUMP TO OUR HANG UP ROUTINE.
0800     A0 20  86      SLOTINIT LDY #$20      ;SLOT 2
0802   BC F1 02  87      STY SLOT
0805   20 4C 0D  88      JSR RESIRO
0808   29 FE 0B  89      JSR HANG1
0808   A9 00  90      LDA #$00      ;CLEAR CARRIER STATUS (RAM VER
SIUN)
080D   85 1F  91      STA CARS1S      ;TO WHATEVER IT IS NOW.
080F   80  92      RTS
0810      93      ;
0810      94      ;
0810      95      ; THE FOLLOWING ROUTINE IS THE OUTPUT
0810      96      ;TO MODEM ROUTINE, SEND A CHARACTER TO
0810      97      ;IT THROUGH THE ACCUMULATOR. IN A
0810      98      ;ACTUAL PROGRAM IT WOULD ALSO BE A GOOD
0810      99      ;IDEA TO CHECK FOR CARRIER BEFORE
0810      100     ;JUMPING TO THIS ROUTINE.
0810   48  101     TTOUT  PHA      ;SAVE THE INPUT
0811   AC F1 02  102     LDY SLOT      ;GET THE SLOT OF THE MODEM
0814   B9 8C 00  103     T10U11  LDA STATUS,Y      ;SEE IF MODEM READY
0817   29 10  104     AND #%200010000      ;TO SEND A CHARACTER OUT
0819   F0 F9  105     BEQ TTOUT1      ;LOOP UNTIL READY
0818      106     ;NOW WE HAVE THE GO-AHEAD FROM THE MODEM
0818      107     ;SO LETS GIVE IT WHAT IT WANTS...
081B   68  108     PLA      ;RECOVER THE ACCUM.
081C   99 8C 00  109     STA OUTDAT,Y      ;SEND IT ON TO THE MODEM.
081F   80  110     RTS
0820      111      ;
0820      112      ; THE FOLLOWING ROUTINE IS THE INPUT
0820      113      ;FROM MODEM ROUTINE, IT WILL RETURN
0820      114      ;A CHARACTER THROUGH THE ACCUM. AGAIN.
0820      115      ;IT WOULD BE A GOOD IDEA IF YOU TESTED
0820      116      ;FOR CARRIER BEFORE CALLING THIS ROUTINE

```

LISA 2.5

CATPACK

```

0820      117 ITIN:
0820 AC F1 02 118 LDY SLOT      ;GET THE SLOT OF THE MODEM
0823 B9 8F C0 119 TTIN1 LDA STATUS,Y   ;CHECK TO SEE IF
0826 29 08 120 AND #X00001000  ;A CHAR WAITING TO BE READ.
0828 F0 F9 121 BEQ TTIN1    ;NO SO KEEP WAITING.
082A      122 ;NOW WE HAVE THE GO-AHEAD FROM THE MODEM, SO LET'S GET IT.
082A A5 1B 123 LDA XMTRAM     ;GET RAM VERSION OF XMITTER
082C 99 BD C0 124 STA XMTRTY,Y   ;CLEAR STATUS PORT
082F B9 BD C0 125 LDA INDAT,Y   ;GET THE CHARACTER
0832 60 126 RTS             ;RETURN WITH A CHAR!
0833      127 ;
0833 128 ;ROUTINE TO CHECK FOR CARRIER
0833 129 ;IF NO CARRIER THEN CHECK FOR
0833 130 ;RING.. AND ATTEMPT TO HOOK UP.
0833 131 ;IF WE AREN'T HOODED UP AFTER ALL
0833 132 ;THAT, THEN WE RETURN WITH THE
0833 133 ;CARRY BIT CLEAR OTHERWISE WE
0833 134 ;RETURN WITH THE CARRY BIT SET TO
0833 135 ;INDICATE THAT WE ARE CONNECTED
0833 136 ;THIS ROUTINE ALSO DISPLAYS PROMPTS
0833 137 ;AS TO WHAT IS GOING ON.
0833 138 ;IT WILL ALSO DETECT A CHANGE IN THE
0833 139 ;STATUS OF THE CARRIER AND HANGUP THE
0833 140 ;PHONE IF THE CARRIER WAS ON AND IS
0833 141 ;NOW OFF.
0833 20 FE 08 142 HANGUP JSR HANG1 ;HANG UP PHONE
0836 20 2B 09 143 JSR PRINTIT   ;TELL THE USER THAT THE FONE H
AS
0839 BD 144 HEX BD          ;HUNG UP
083A C1 D0 D0 145 ASC "APPLE-CAT:HUNG UP"

083D CC C5 AD
0840 C3 C1 D4
0843 BA CB D5
0846 CE C7 A0
0849 D5 D0
084B BD 00 146 HEX BD00
084D 4C 62 08 147 JMP CHKRN
0850 38 140 CARFND SEC      ;CONTINUE ON...
0851 60 149 RTS             ;SET CARRY TO INDICATE THAT
0852 AC F1 02 150 CARCHK LDY SLOT   ;WE HAVE A CARRIER AND RETURN
0855 B9 80 C0 151 LDA SWBYT,Y   ;GET THE CARD SLOT
0858 29 20 152 AND #X00100000 ;GET CARRIER RYTE
085A D0 F4 153 BNE CARFND    ;MASK CARRIER BIT
085C A5 1F 154 LDA CARSTS   ;WE HAVE A CARRIER
085E 29 20 155 AND #X00100000 ;GET OLD STATUS
0860 D0 D1 156 RNE HANGUP  ;SEE IF IT WAS ON
0861      157 ;IT WAS ON SO WE HANG UP THE P
HONE
0862 B9 8D C0 157 CHCRNG LDA ACBYT,Y   ;IT WAS ON SO WE HANG UP THE P
0865 29 01 158 AND #X00000001 ;NO CARRIER, SO CHECK FOR RING
0867 F0 6E 159 BEQ NOCAR    ;MASK RING BIT.
0869 20 2B 09 160 JSR PRINTIT   ;PHONE NOT RINGING.
086C BD 161 HEX BD
086D 41 50 50 162 ASC "APPLE-CAT:RING.."
0870 4C 45 2D
0873 43 41 54
0876 3A 52 49
0879 4E 47 2E
087C 2E

```

21

LISA 2.5

CATPACK

```

087D BD 00 163 HEX BD00
087F B9 8D C0 164 FINRNG LDA ACBYT,Y   ;LET IT FINISH RINGING
0882 29 01 165 AND #X00000001 ;MASK RING BIT
0884 D0 F9 166 BNE FINRNG ;STILL RINGING
0886 167 ;PHONE IS DONE RINGING, NOW WE WILL
0886 168 ;WAIT FOR A CARRIER TO COME IN.
0886 20 5F 09 169 JSR INIT    ;TURN ON MODEM
0889 170 ;----- ;GIVE OTHER SIDE 20 SECS TO DET
0890 171 ;CARRIER GOING.
0889 20 2B 09 173 JSR PRINTIT
0890 BD 174 HEX BD
0891 41 50 50 175 ASC "APPLE-CAT:AWAIT CARRIER"

0890 4C 45 2D
0893 43 41 54
0896 3A 41 57
0899 41 49 54
089C 20 43 41
090F 52 52 49
08A2 45 52
08A4 00 176 HEX 00
08A5 A2 28 177 LDX #140
08A7 AC F1 02 178 ANSLP1 LDY SLDT
08A8 B9 8D C0 179 LDA SWBYT,Y   ;GET CARRIER BYTE.
08AD 29 20 180 AND #X00100000 ;MASK CARRIER BIT.
08AF D0 2C 181 BNE PRTCAR ;WE HAVE A CARRIER!
08B1 A9 05 182 LDA #1'S   ;.5 SEC
08B3 20 19 09 183 JSR WAIT1 ;WAIT ROUTINE
08B6 CA 184 DEX
08B7 D0 EE 185 BNE ANSLP1 ;STILL HAVE TIME LEFT
08B9 20 2B 09 186 JSR PRINTIT
08B8 BD BD 188 HEX BD8D
08B6 41 50 50 189 ASC "APPLE-CAT:NO CARRIER"

08C1 4C 45 2D
08C4 43 41 54
08C7 3A 4E 4F
08CA 20 43 41
08CD 52 52 49
08D0 45 52
08D2 BD 00 190 HEX BD00
08D4 20 33 08 191 JSR HANGUP ;HANG UP FONE.
08D7 A9 00 192 NOCAR LDA #00 ;NO CARRIER
08D9 85 1F 193 STA CARSTS
08DB 18 194 CLC ;CLEAR CARRY TO INDICATE NO CA
RRIE
08DC 60 195 RTS ;RETURN TO CALLING ROUTINE
08DD 20 33 09 196 PRTCAR JSR PRINTIT
08E0 BD BD 197 HEX BD8D
08E2 41 50 50 198 ASC "APPLE-CAT:CONNECTED"
08E5 4C 45 2D
08EB 43 41 54
08ED 3A 43 4F
08EE 4E 4E 45
08F1 43 54 45
08F4 44
08F5 BD 00 199 HEX BD00

```

22

```

08F7 A9 20    200    LDA #%00100000 ;SET CARRIER STATUS TO ON
08F9 85 1F    201    STA CARS1
08FB AC 50 0B  202    JMP CARFND ;ACKNOWLEDGE THAT WE HAVE A CA
RRIE
08FE AC F1 02  203    HANG1 LDY SL01
0901 78    204    SEI ;TURN OFF INTERRUPTS
0902 A9 00    205    LDA #00 ;PUT THE PHONE ON THE RING
0904 99 82 C0  206    STA SIBYT,Y
0907 A9 01    207    LDA #%00000001 ;MIC & SPK
0909 99 81 C0  208    STA SCIBYT,Y ;OFF.
090C A9 0F    209    LDA #$0F ;MAKE SURE THAT WE DON'T
090E 99 83 C0  210    STA BSIBYT,Y ;GET ANY INTERRUPTS FROM
0911 A9 1F    211    LDA #$$1F ;BSR & TURN OFF XMITTER
0913 99 8D C0  212    STA XMIBYT,Y
0916 85 1B    213    STA XMTRAM
0918 80    214    RTS ;RETURN
0919 215    ;----- ;ENTER THIS ROUTINE WITH # OF LOOMS
0919 216    ;----- ;TIME PERIODS IN THE ACCUM.
0919 86 1C    218    WAIT1 STX TEMPX ;SAVE X
091B 84 1D    219    STY TEMPY ;SAVE Y
091D A8      220    TAY ;MOVE COUNT TO Y REG
091E A9 C0    221    WAIT1A LDA #'195 ;DO 100MS WAIT
0920 20 A8 FC  222    JSR WAIT ;DELAY
0923 88    223    DEY
0924 D0 F8    224    BNE WAIT1A ;Y=0? NO-GO BACK
0926 A6 1C    225    LDX TEMPX ;RESTORE X
0928 A4 1D    226    LDY TEMPY ;RESTORE Y
092A 80    227    RTS ;RETURN
092B 228    ;----- ;ROUTINE TO PRINT OUT A STRING
092B 229    ;----- ;OF CHARACTERS ENDING WITH A $00
092B 230    ;----- ;PRINT1 STY TEMPY ;SAVE Y
092B 94 1D    231    PRINT1 STX TEMPX ;SAVE X
092D 85 1C    232    PLA
092F 88    233    PLA C3
0930 85 04    234    STA C3
0932 68    235    PLA
0933 85 05    236    STA C3+$1
0935 A0 01    237    LDY #1
0937 B1 04    238    PRINT2 LDA (C3),Y
0939 C9 00    239    CMP #00
0938 F0 08    240    BEQ QUIT
093D 09 80    241    DRA #480 ;SET HIGH BIT
093F 20 F0 FD  242    JSR COUT1
0942 20 58 09  243    JSR UPDATE
0945 40 37 09  244    JMP PRINT2
0948 20 58 09  245    QUIT JSR UPDATE
094B A5 05    246    LDA C3+$1
094D 48      247    PHA
094E A5 04    248    LDA C3
0950 48      249    PHA ;RESTORED THE STACK
0951 A6 1C    250    LDX TEMPX ;RESTORE X
0953 A4 1D    251    LDY TEMPY ;RESTORE Y
0955 A9 01    252    LDA #1 ;LEAVE WITH ACCUM=1 ALWAYS
0957 60      253    RTS
0958 E6 04    254    UPDATE INC C3
095A D0 02    255    BNE DUITR

```

```

095C E6 05    256    INC C3+$1
095E 60    257    DUITR RTS
095F 258    ;----- ;FOLLOWING IS A TYPICAL ROUTINE THAT
095F 259    ;----- ;YOU WOULD JUMP TO WHEN YOU WANTED TO
095F 260    ;----- ;INITIALIZE THE MODEM FOR DATA MODE,
095F 261    ;----- ;MAKE SURE NO INTERRUPTS OCCUR.
095F 262    ;----- ;GET SLOT NUMBER
095F 263    INIT SE1 ;MAKE SURE NO INTERRUPTS OCCUR.
095G AC F1 02  264    LDY SL01 ;GET SLOT NUMBER
0963 A9 01    265    LDA #%00000001 ;TURN OFF MIC & SPEAKER.
0965 99 81 C0  266    STA SCIBYT,Y ;OFF.
0968 A9 02    267    LDA #%00000010 ;TELL MODEM TO
096A 99 82 C0  268    STA SIBYT,Y ;PICK UP THE PHONE.
096D A9 06    269    LDA #%000000110 ;DISABLE BSR & 30HZ INTERRUPTS.
0972 A9 64    270    STA BSIBYT,Y ;;
0974 99 89 C0  271    LDA #$$4 ;ANS 103
0977 A9 07    272    STA RECBYT,Y ;MODE FOR THE RECEIVER.
0979 99 8A C0  273    LDA #%00000111 ;8 DATA + 1.5 STOP
097E 99 8B C0  274    STA COM,Y ;BITS, NO PARITY.
097F A9 22    275    LDA #$22 ;SET SPEED TO 300 BAUD
097E 99 8B C0  276    STA SPDBYT,Y ;XMIT & RECV
0981 A9 05    277    LDA #%00000101 ;XMIT & RECV WITH NU
0983 99 8C C0  278    STA COM,Y ;INTERUPTS
0986 A9 10    279    LDA #$10 ;ANS 103
0988 99 8D C0  280    STA XMIBYT,Y ;MODE FOR THE TRANSMITTER.
0988 85 10    281    STA XMTRAM ;RAM VERSION OF SAME.
098D A9 00    282    LDA #$00 ;DIRECT DATA TO AND
098F 99 8F C0  283    STA DEBVT,Y ;FROM MAIN MODEM.
0992 60    284    RTS
0993 285    ;----- ;ROUTINE TO INITIALIZE MODEM FOR VOICE
0993 286    ;----- ;MODE.
0993 AC F1 02  288    INITV LDY SL01 ;GET MODEMS SLOT
0996 A9 03    289    LDA #%00000011 ;TURN ON SWITCH HOOK
0998 99 82 C0  290    STA SIBYT,Y ;AND ENABLE DV.
0998 A9 02    291    LDA #%00000010 ;TURN ON SPK & MIC.
099D 99 81 C0  292    STA SCIBYT,Y
09A0 A9 1F    293    LDA #%00011111 ;TURN OFF XMITTER.
09A2 99 80 C0  294    STA XMIBYT,Y
09A5 85 1B    295    STA XMTRAM
09A7 60    296    RTS
09AB 297    ;----- ;ROUTINE TO READ A TOUCHTONE FROM THE
09AB 298    ;----- ;MODEM. YOU MUST HAVE TOUCH TONE OPTION
09AB 299    ;----- ;TO USE THIS ROUTINE. ALSO MAKE SURE
09AB 300    ;----- ;THE SWITCH HOOK IS ENABLED TO RECEIVE
09AB 301    ;----- ;A TOUCHTONE. EG. SEND A #%00000111 TO
09AB 302    ;----- ;THE ACCUMULATOR.
09AB 303    ;----- ;THE CHAR WILL BE RETURNED THROUGH
09AB 304    ;----- ;THE ACCUMULATOR.
09AB 305    ;----- ;THE ACCUMULATOR.
09AB I9 80 C0  306    RECTIONE LDA SIBYT,Y ;IS THERE A TONE
09AB 29 80    307    AND #$80 ;TONE PRESENT?
09AD F0 F9    308    BEQ RECTIONE ;LOOP TILL WE GET ONE
09AF B9 81 C0  309    LDA TONBYT,Y ;GET TONE DATA
09B0 29 0F    310    AND #%00000111 ;LOOK AT LO-NIBBLE ONLY.
09B4 AA    311    FAX ;TRANSFER TO THE X REG

```

LISA 2.5

## CATPACK

```

09B5 DD C5 09 312 LDA TTABLE,X ;GET THE CHAR FROM THE TABLE
09B8 48 313 PHA ;SAVE THE CHAR FOR A MOMENT
09B9 A9 02 314 LDA #2 ;CLEAR DV BY
09B8 99 B2 C0 315 STA SHBYT,Y ;STROKING CLRDV
09B8 A9 03 316 LDA #3 ;ON AND THEN
09C0 99 B2 C0 317 STA SHBYT,Y ;OFF AGAIN
09C3 6B 318 PLA ;RECOVER CHAR.
09C4 60 319 RTS
09C5 R0 R1 B2 320 TTITLE ASC "01234567890*#ABCD"
09C8 B3 B4 B5
09CB B6 B7 B8
09CE B9 B0 AA
09D1 A3 C1 C2
09D4 C3 C4
09D6 321 ;-----:PRINTER OUTPUT ROUTINE; THIS ROUTINE
09D6 322 ;ASSUMES THAT YOU HAVE ALREADY CORRECTLY
09D6 323 ;SET UP THE PRINTER INTERFACE (SPEED,
09D6 324 ;PARITY,BITS, ETC) BEFORE CALLING THIS
09D6 325 ;ROUTINE.
09D6 326 ;SEND THE CHAR TO BE XMITTED THROUGH
09D6 327 ;THE ACCUMULATOR.
09D6 328 PRNTOUT LDY SLOT
09D6 AC F1 02 329 LDA ACBYT,Y ;GET BYTE WITH HANDSHAKE BIT.
09D9 B9 BD C0 330 AND #200001000 ;MASK OUT BIT
09DC 29 08 331 BEQ PRNTOUT ;LOOP TILL READY.
09DE F0 F6 332 JSR TTOUT ;BRANCH TO THE CHAR OUT SUBROUTINE
09E0 20 10 0B 333
TINE
09E3 60 334 RTS ;RETURN
09E4 335 ;-----:FOLLOWING IS A COMPLETE DIALING
09E4 336 ;PACKAGE; IT WILL HANDLE BOTH TOUCH
09E4 337 ;PACAGE, IT WILL HANDLE BOTH TOUCH
09E4 338 ;TONE AS WELL AS PULSE DIALING.
09E4 339 ;THIS ROUTINE IS FREE STANDING, SO
09E4 340 ;THERE IS NO NEED TO SEND ANY CHARS
09E4 341 ;TO IT, HOWEVER YOU MAY WANT TO
09E4 342 ;BY-PASS THE MENU ROUTINES AND JUMP
09E4 343 ;DIRECTLY INTO THE DIALING PORTION.
09E4 344 ;
09E4 345 ;THE DIALING ROUTINE WILL RETURN
09E4 346 ;A #01 IN THE ACCUMULATOR IF THE
09E4 347 ;DIALING WAS ABORTED, IF IT WAS
09E4 348 ;COMPLETED IT WILL RETURN A #00
09E4 349 ;NOTE: THIS ROUTINE WILL RETURN WITH
09E4 350 ;INTERRUPTS OFF, SO YOU MAY NEED TO
09E4 351 ;TURN THEM BACK ON AFTER CALLING THIS
09E4 352 ;IF YOU ARE USING INTERRUPTS
09E4 353 ;
09E4 8A 354 IRDROUT TXA
09E5 48 355 PHA ;SAVE X
09E6 98 356 TYA
09E7 48 357 PHA ;SAVE Y
09E8 AC F1 02 358 LDY SLOT ;GET SLOT
09E8 B9 B1 C0 359 LDA TONBYT,Y ;GET TONEBYTE
09EE 29 10 360 AND #200010000 ;MASK OUT 30 HZ BIT
09F0 D0 17 361 BNE RESTDR ;RETURN
09F2 A9 04 362 LDA #200000100 ;RESET AND RE-

```



LISA 2.5

## CATPACK

```

09F4 99 B3 C0 363 STA BSRBT,Y ;ENABLE BSR
09F7 A9 00 364 LDA #200000000 ;IN0'S
09F9 99 B2 C0 365 STA BSRBT,Y ;GET COUNT
09FC A5 11 366 LDA COUNT ;DONE
09EE F0 04 367 BNE OFF ;DECOUNT
09A0 C6 11 368 DEC COUNT ;DECREMENT TIMER
09A2 D0 05 369 BNE RESTOR ;RETURN
09A4 A9 04 370 OFF LDA #200000100 ;TURN OFF 30HZ
09A6 99 B3 C0 371 STA BSRBT,Y ;IN0'S
09A9 6B 372 RESTDR PLA ;FULL Y
09A0 A8 373 TAY ;RESTORE Y
09A8 6B 374 PLA ;FULL X
09A0 AA 375 TAX ;RESTORE X
09AD A5 45 376 LDA $45 ;RECOVER ACCUMULATOR
09AF 40 377 RTI ;RETURN FROM INTERRUPT
09A1 A9 E4 378 SETIRO LDA #IRQROUT ;GET LO BYTE LOCATION
09A2 B9 FE 03 379 STA IRQ ;SAVE LOW IRQ BYTE
09A5 A9 09 380 LDA /IRQROUT ;GET HI BYTE
09A7 BD FF 03 381 STA IRQ+1 ;SAVE HI BYTE
09A1 60 382 RTS ;GO BACK
09A1B 383 ;-----: HERE IS THE ACTUAL DIALING ROUTINE
09A1B 384 ;SO YOU WOULD BRANCH HERE TO DIAL A
09A1B 385 ;NUMBER.
09A1B 386 ;-----: NUMBER.
09A1B 387 ;-----: NUMBER.
09A1B 388 DIALER JSR SETIRO ;SET UP THE INTERRUPT ROUTINE
09A1E 20 00 08 389 JSR SLOTINIT ;TO MAKE IT FREE STANDING
09A21 20 58 FC 390 JSR HOME ;CLEAR SCREEN
09A24 20 28 09 391 JSR PRINTIT ;PRINT WHAT FOLLOWS
09A27 00 00 392 BYT $D,$D
09A29 BC C4 BE 393 BYT "<D> DIAL A NUMBER", $D
09A2C A0 A0 C4
09A2F C9 C1 CC
09A32 A0 C1 A0
09A35 CE D5 CD
09A38 C2 C5 D2
09A3B OD
09A3C IC D2 BE 394 BYT "<D> RE-DIAL LAST NUMBER", $D
09A3F A0 A0 D2
09A42 C5 AD C4
09A45 C9 C1 CC
09A48 A0 CC C1
09A4B D5 D4 A0
09A4E CE D5 CD
09A51 C2 C5 D2
09A54 OD
09A55 BC C5 D3 395 BYT "<ESC> ABORT DIALING", $D,$D
09A58 C3 BE A0
09A5B C1 C2 CF
09A5E D2 D4 A0
09A61 C4 C9 C1
09A64 CC C9 CE
09A67 C7 OD OD 396 BYT "CHOICE?", 0
09A6A C3 CB CF
09A6D C9 C3 C5
09A70 BF 00

```



LISA 2.5

CATPACK

```

0A72 20 C9 0E 397      JSR COMND1      :COMMAND PROCESSOR
0A75 44 398          BYT "D"
0A76 86 0A 399          ADR DIAL
0A78 52 400          BY1 "R"
0A79 8C 0A 401          ADR REDIAL
0A7B 1B 402          BYT "#1B"
0A7C 82 0A 403          ADR ABORT
0A7E 1B 404          BYT ";"      ;END OF DATA
0A7F 4C 1B 0A 405      JMP DIALER      ;LOOP BACK
0A82 20 4C 0D 406      ABORT      JSR RESIRD
0A85 60 407          RTS
0A86 20 00 0A 408      DIAL      JSR DIALSB
0A89 4C 8F 0A 409      JMP DIALDT
0ABC 20 CA 0B 410      REDIAL      JSR REDLSB
0ABF AC F1 02 411      DIALDT      LDY SLOT
0A92 A9 21 412          LDA #$21      ;PUT RECEIVER IN 103 ORIG
0A94 99 89 C0 413          STA RECBYT,Y
0A97 4C 82 0A 414      JMP ABORT
0A9A 415          ;----- ROUTINE TO GET TELEPHONE # IN
0A9A 416          ;ROUTINE TO GET TELEPHONE # IN
0A9A 417          ;DIGIT BUFFER ($250-$389)
0A9A A9 8A 418          GETNO1      LDA #$BA      :: PROMPT
0A9C 85 33 419          STA PROMPT      ;PUT IN MONITR LOCATION
0A9E 20 6A FD 420          JSR GETLN      ;GET A LINE OF TEXT
0AA1 A9 00 421          LDA #0      ;ZERO THE
0AA3 85 10 422          STA C7      ;INDEX COUNTER
0AA5 A6 10 423          GETNO1      LDX C7      ;GET THE VALUE OF THE INDEX CO
UNT
0AA7 E0 28 424          CPX #$38      ;IF TOO LONG THEN TRUNCATE
0A99 F0 0F 425          BEQ OUT13
0AAB FD 00 02 426          LDA KEYBUF,X      ;GET THE CHAR IN THE BUFFER
0AAE C9 8D 427          CMP #$8D      ;IS IT C/R
0AB0 F0 08 428          BEQ OUT13      ;YES RETURN TO CALLER
0AB2 9D 50 03 429          STA DIGBUF,X      ;STORE IN DIGIT BUFFER
0AB5 E6 10 430          INC C7      ;INC COUNTER
0AB7 4C A5 0A 431          JMP GETNO1      ;LOOP FOR MORE
0ABA A9 8D 432          OUT13      LDA #$8D
0ABC 9D 50 03 433          STA DIGBUF,X      ;PUT RETURN IN LAST LOC
0ABF 60 434          RTS
0AC0 435          ;----- ROUTINE TO CLEAR SCREEN
0AC0 20 58 FC 436          DIALSB      JSR HOME      ;CLEAR SCREEN
0AC3 20 2B 09 437          JSR PRINFIT
0AC6 BC B0 AD 438          BYT "(0-9,*,#) DIGITS TO DIAL",#D
0AC9 B9 AC AA
0ACC AC A3 DE
0ACF A0 C4 C9
0AD2 C7 C9 D4
0AD5 D3 A0 D4
0AD8 CF A0 C4
0ADB C9 C1 CC
0AE1 0D
0ADF EC D4 BE 439          BYT "ST2"      TOUCH TONE SELECT",#D
0AE2 A0 A0 A0
0AE5 A0 A0 A0
0AE8 A0 D4 CF
0AE9 D5 C3 C8
0AE6 A0 D4 CF
0AF1 CE C5 A0

```

27

LISA 2.5

CATPACK

```

0AF4 D3 C5 CC
0AF7 C5 C3 D4
0AFA 0D
0AFB BC D0 BE 440      BYT "<P>"      PULSE DIAL SELECT",#D
0AFA A0 A0 A0
0B01 A0 A0 A0
0B04 A0 D0 D5
0B07 CC D3 C5
0B0A A0 C4 C9
0B0D C1 CC A0
0B10 D3 C5 CC
0B13 C5 C3 D4
0B16 0D
0B17 BC AB BE 441      BYT "<+>"      PAUSE FOR DIALTONE",#D
0B1A A0 A0 A0
0B1D A0 A0 A0
0B20 A0 D0 C1
0B22 D5 D3 C5
0B26 A0 C6 CF
0B29 D3 A0 C4
0B2C C9 C1 CC
0B2F D4 CF CE
0B32 C5 0D
0B34 EC C0 BE 442      BYT "<2>"      PAUSE FOR 2 SECONDS",#D, #D
0B37 A0 A0 A0
0B3A A0 A0 A0
0B3D A0 D0 C1
0B40 D5 D3 C5
0B43 A0 C6 CF
0B46 D2 A0 B2
0B49 A0 D3 CF
0B4C C3 EF CE
0B4F C4 D3 OD
0B52 0D
0B53 CE LF D4 443      BYT "NOTE: P> & <1>, IF USED MUST BE THE",#D
0B56 C5 BA A1
0B59 RC D0 RE
0B5C A0 A6 A0
0B5F RC D4 BE
0B62 A0 C9 C6
0B65 A0 D5 D3
0B68 C5 C4 A0
0B6B CD D5 D3
0B6E D4 A0 C2
0D71 C5 A0 D4
0B74 CB C5 OD
0B77 C6 C9 D2 444      BYT "FIRST LETTERS IN THE NUMBER",#D
0B7A D3 D4 A0
0B7D CC C5 D4
0B80 D4 C5 D2
0B83 D3 A0 C9
0B86 CE A0 D4
0B89 CB C5 A0
0B8C CE D5 CD
0B8F C2 C5 D2
0B92 0D

```

28



LISA 2.5

CATPACI

```

OB93 CE D5 CD 445 BYT "NUMBER TO DIAL",0
OB96 C2 C5 D2
OB99 A0 D4 CF
OB9C A0 C4 C9
OB9F C1 CC 00
OBAA 20 9A 0A 446 JSR GETNO :GET #
OBAB A2 00 447 LDX #0 :INIT INDEX
OBAB BD 50 03 448 LDA DIGBUF,X :GET CHAR
OBAA C9 BD 449 CMP #$BD :IS IT <CR>
OBAC D0 03 450 BNE REDLSB1 :NO-SKIP NEXT LINE.
OBAE 4C D1 0C 451 JMP DLAB :YES-JUMP TO ABORT
OBBI C9 D4 452 REDLSB1 CMP #"T"
OBBD DO 0A 453 BNE REDLSB2
OBBS 4B 454 FMA
OBBA A9 00 455 LDA #00
OBEB BD 0C 03 456 STA DIALSW
OBBL 6B 457 PLA
OBBC 4C CA 0B 458 JMP REDLSB
OBBF C9 D0 459 REDLSB2 CMP #"P"
OBCC D0 07 460 BNE REDLSB
OBCE 4B 461 FMA
OBCA A9 01 462 LDA #01
OBCE BD 0C 03 463 STA DIALSW
OBCH 6B 464 PLA
OBGA 20 4C 0D 465 REDLSB JSR RESIRO :TURN OFF ACTA
OBGD 20 5B FC 466 JSR HOME :CLEAR SCREEN
OBDO 20 2B 09 467 JSR PRINTIT
OBDS 20 20 20 468 BYT "ESCAPE TO ABORT",$,D,$D
29
OB6E 3C 45 53
OB99 43 3E 20
OBDC 54 4F 20
OBDF 41 42 4F
OBE2 52 54 0D
OBES 0D
OBE6 0D 0D 0D 469 BYT $D,$D,$D,"DIALING",0
OBEC 0D 44 49
OBEF 4E 47 3A
OBF2 00
OBF3 AC F1 02 470 LDY SLOT :GET CARD INDEX
OBF6 A9 03 471 LDA #%000000011 :TURN OFF
OBF8 99 81 C0 472 STA SDOUY1,Y :MICROPHONE
OBF9 A9 7E 473 LDA #$7E :SET RECEIVER
OBF9 99 89 C0 474 STA RECVYT,Y :TO DIAL-TONE DETECT MODE
OC00 A9 1F 475 LDA #$1F :SET XMITTER
OC02 99 8D C0 476 STA XMTBYT,Y :TO OFF
OC05 85 1B 477 STA XMTRAM :STORE IN RAM VERS. OF XMTBYT.
OC07 :-----
OC07 A9 00 479 LDA #0 :TURN
OC09 99 82 C0 480 STA SHBYT,Y :OFF SH
OC0C :-----
OC0C A9 1E 482 LDA #30 :TURN OFF SH FOR
OC0E 20 19 09 483 JSR WAIT1 :3 SEC.
OC11 A9 03 484 LDA #3 :TURN ON
OC13 99 82 C0 485 STA SHBYT,Y :SH AGAIN

```



LISA 2.5

CATPACI

```

OC16 486 :-----
OC16 487 :DIAL TONE DETECTION ROUTINE
OC16 20 BA 0E 488 DIAL1A JSR RDKEY :LOOK AT KEYBOARD
OC19 90 07 489 BCC DIAL1B :NO CHAR SO SKIP TO DIAL1A
OC1B C9 1B 490 CMP #$1B :IS CHAR <ESC>
OC1D D0 03 491 BNE DIAL1B :NO-JUMP NEXT LINE.
OC1F 4C D1 0C 492 JMP DLAB :YES-ABORT
OC22 99 80 C0 493 DIAL1B LDA SWBYT,Y :GET BYTE WITH CARRIER
OC25 29 10 494 AND #Z00010000 :MASK ZERO CROSSING BIT
OC27 DO ED 495 BNE DIAL1A :IF NOT DIAL TONE THEN LOOP
OC29 496 :OTHERWISE MAKE SURE DIAL TONE IS
OC29 497 :AROUND FOR 1 SEC.
OC29 A9 1D 498 LDA #29 :SET UP
OC2B 20 21 0D 499 JSR COUNTM :1 SEC. WAIT + COUNT 0-CROSSI
NGS.
OC2E C9 01 500 CMP #1 :1 CROSSING MEANS DIAL TONE.
OC30 DO E4 501 BNE DIAL1A :START OVER AGAIN
OC32 502 :-----
OC32 A2 00 503 LDX #0 :INIT INDEX COUNTER.
OC34 BD 50 03 504 DIAL1 :LDA DIGBUF,X :GET 1ST CHAR.
OC37 C9 BD 505 CMP #$BD :IS IT <CR>
OC39 F0 70 506 BEU DIAL0V :YES-GO TO END ROUTINE.
OC3B 49 507 DIAL2 FMA :SAVE CHAR
OC3C 20 ED FD 508 JSR COU1 :PRINT CHAR
OC3F 6B 509 PLA :RECOVER CHAR
OC40 EB 510 INX :INCREMENT CHAR. COUNTER
OC41 4B 511 FMA :SAVE IN STACK
OC42 20 B4 0E 512 JSR RDKEY :LOOK AT KEYBOARD
OC45 90 07 513 BCC DIAL3 :NO CHAR SO SKIP TO DIAL3
OC47 C9 1B 514 CMP #$1B :IS CHAR <ESC>
OC49 D0 03 515 BNE DIAL3 :NO-SKIP NEXT CODE.
OC4B 4C D0 0C 516 JMP DIALAB :YES TO ABORT MESSAGE.
OC4E 6B 517 DIAL3 PLA :RESTORE A
OC4F C9 C0 518 CMP #$C0 :IS IT A 'P'
OC51 DO 07 519 BNE PLUS :NO-SO SKIP TO PLUS
OC53 4B 520 FMA :SAVE CHAR IN STACK
OC54 A9 14 521 LDA #20 :WAIT FOR
OC56 20 19 09 522 JSR WAIT1 :2 SEC.
OC59 6B 523 PLA :RECOVER CHAR
OC5A C9 AB 524 PLUS CMP #$AB :IS IT '+'(PAUSE)
OC5C DO 1B 525 BNE DIAL3 :NO-ROTO DIAL3A
OC5E 526 :-----
OC5E 527 :DIAL TONE DETECTION ROUTINE FOR '+' :SAVE CHAR IN STACK
OC5E 4B 528 FMA :LOOK AT KEYBOARD
OC5F 20 BA 0E 529 DTD1 JSR RDKEY :NO CHAR SO SKIP TO DIAL1A
OC62 90 04 530 BCC DTD2 :IS CHAR <ESC>
OC64 C9 1B 531 CMP #$1B :YES-ABORT
OC66 F0 6B 532 BEU DIALAB :GET BYTE WITH CARRIER
OC68 B9 80 C0 533 DTD2 LDA SWBYT,Y :MASK ZERO CROSSING BIT
OC6B 29 10 534 AND #Z00010000 :IF NOT DIAL TONE THEN LOOP
OC6D DO F0 535 BNE DTD1
OC6F 536 :OTHERWISE MAKE SURE DIAL TONE IS
OC6F A9 1D 538 LDA #29 :SET UP
OC71 20 21 0D 539 JSR COUNTM :1 SEC. WAIT + COUNT 0-CROSSI
NGS.
OC74 C9 01 540 CMP #1 :1 CROSSING MEANS DIAL TONE.
OC76 DO E7 541 BNE DTD1 :START OVER AGAIN

```

30

LISA 2.5

CATPACK

```

OC7B 6B      542 PLA          ;RECOVER CHAR
OC79          543 ;-----
OC79 C9 AA    544 DIAL3A   CMP #$AA   ;IS IT '*' ?
OC7B D0 04    545 BNE D$A1   ;NO-SKIP AHEAD
OC7D A9 0A    546 LDA #$A   ;YES-PUT $A IN ACCUM
OC7F D0 18    547 BNE DIAL3B ; AND SEND IT
OC81 C9 A3    548 D$A1   CMP #$A3  ;IS IT '#' ?
OC83 D0 04    549 BNE D$A2   ;NO-SKIP AHEAD
OC85 A9 0B    550 LDA #$B   ;YES-PUT $B IN ACCUM
OC87 D0 10    551 BNE DIAL3B ; AND SEND IT
OC89          552 ;-----
OC89 3B      553 D$A2   SEC          ;PREPARE FOR SUBTRACT WITHOUT
CARR
OC8A E9 B0    554 SRC #$B0
OC8C 30 1A    555 BMI DIALS
OC8E C9 0A    556 CMP #$A
OC90 10 16    557 BPL DIALS
OC92 4B      558 PHA          ;GO GET NEXT # (TOO LARGE)
OC93 AD 0C 03  559 LDA DIALSW ;SAVE # ON STACK
OC96 D0 07    560 BNE DIAL4
OC98 6B      561 PLA          ;IS TOUCH-TONE ENABLED(=0)
OC99 20 40 0E  562 DIAL3B   JSR TONE   ;SEND THE TOUCH-TONE
OC9C 4C 34 0C  563 JMP DIAL1  ;GO GET NEXT CHAR
OC9F A9 0B    564 DIAL4   LDA #B   ;BETWEEN-DIGIT
OC81 30 19 09  565 JSR WAIT1  ;WAIT OF 800 MS
OC84 6B      566 PLA          ;RECOVER A REG
OC85 20 F4 0C  567 JSR PULSIT ;GO DIAL #
OC8B 4C 34 0C  568 DIAL5   JMP DIAL1  ;GO GET NEXT #
OCAB          569 ;-----
OCAB 20 2B 09  570 DIAL0V  JSR PRINT1  ;DIALING-OVER ENDING
OCAE 0D 0D 44  571 BYT $D,$D,'DIALING COMPLETED',0

31

OC22 AC F1 02 572 LDY SLOT    ;GET CARD INDEX
OC55 A9 02    573 LDA #%00000010 ;TURN ON MIC
OC57 99 81 C0  574 STA SQUBT,Y ;AND SPKR
OCCA 20 4C 0D  575 JSR RESIRO  ;ASSURE INTERRUPTS OFF
OCDD A9 00    576 LDA #0   ;SIGNAL DIALING COMPLETED
OCDF 60      577 RIS          ;RETURN FROM DIALING ROUTINE.
OCDO 6B      578 DIALAB   PLA          ;RECOVER CHAR
OCDI AC F1 02 579 DLAB   LDY SLOT    ;GET CARD INDEX
OCD4 A9 00    580 LDA #0   ;TURN OFF
OCDE 99 82 C0  581 STA SQUBT,Y ;SWITCH HOOK
OCDF A9 01    582 LDA #%00000000 ;TURN OFF MIC
OCDB 99 81 C0  583 STA SQUBT,Y ;AND SPKR
OCDE 20 1B 09  584 JSR PRINT1
OC81 0D 0D 44  585 BYT $D,$D,'DIALING ABORTED',0
OC84 49 41 4C
OC87 49 4E 47
OC8A 20 41 42
OC8D 4F 52 54
OC8F 45 44 00
OCF3 60      586 RTS          ;RETURN FROM DIALING ROUTINE.

```

LISA 2.5

CATPACK

```

OCF4          587 ;ACCUM. HAS $01 TO INDICATE ABORT.
OCF4          588 ;-----
OCF4 B6 1C    589 PULSIT    STX TEMPX   ;SAVE X AND
OCF6 84 1D    590 STY TEMPY   ;Y REGS
OCFB C9 00    591 CMP #0   ;IS # 0 ?
OCFA D0 02    592 BNE PULSE1 ;NO-SKIP NEXT STEP
OCFC A9 0A    593 LDA #10  ;YES-MAKE IT 10
OCFE AA      594 PULSE1   TAX          ;TRANSFER A TO X
OCFF AC F1 02 595 PULSE2   LDY SLOT    ;GET CARD INDEX
OD02 A9 00    596 LDA #0   ;TURN
OD04 99 82 C0  597 STA SBYT,Y ;OFF SH
OD07 A9 96    598 LDA #150 ;FOR
OD09 20 AB FC 599 JSR WAIT  ;60 MS.
OD0C A9 03    600 LDA #3   ;TURN ON
OD0E 99 82 C0  601 STA SBYT,Y ;SH AGAIN
OD11 CA      602 DEX          ;DECREMENT X
OD12 F0 08    603 BEQ PULSE3 ;IF X=0 THEN WE ARE DONE
OD14 A9 7B    604 LDA #123 ;WAIT FOR
OD16 20 AB FC 605 JSR WAIT  ;40 MS.
OD19 4C FF 0C 606 JMP PULSE2 ;DO IT AGAIN
OD1C A6 1C    607 PULSE3   LDX TEMPX  ;RESTORE X AND
OD1E A4 1D    608 LDY TEMPY  ;Y REGS.
OD20 60      609 RTS       ;RETURN

OD21          610 ;-----
OD21          611 COUNT1   ;IN THE ACCUMULATOR.
OD21          612 ;THE # OF ZERO CROSSINGS IS RETURNED IN THE ACCUM.
OD21          613 STA COUNT  ;PUT ACCUM IN COUNT REGISTER
OD21 85 11    614 TYA          ;SAVE Y-REG
OD23 98      615 LDA #0   ;IN STACK
OD24 4B      616 PHA          ;GET CARD INDEX
OD25 AC F1 02 617 LDY SLOT    ;TURN ON
OD28 A9 00    618 LDA #0   ;ZERO TRANSITION COUNTER
OD2A 85 02    619 STA CROSS  ;30 Hz IRQ'S
OD2C 99 83 C0  620 STA BSBYT,Y ;INITIALIZE
OD2F A9 10    621 LDA #%200010000 ;ZERO TEMP LOCATION TO 1.
OD31 85 F6    622 STA ZCTMP ;ENABLE IRQS
OD33 5B      623 CLI          ;IS IRQ DELAY ROUTINE DONE?
OD34 A5 11    624 COUNT1   LDA COUNT  ;YES-EXIT ROUTINE
OD36 F0 0F    625 BEQ COUNT3 ;GET SWITCH BYTE
OD39 B9 80 C0  626 COUNT2   LDA SWBYT,Y ;MASK ZERO CROSSING BIT
OD3B 29 10    627 AND #%00010000 ;IS POLARITY SAME AS BEFORE.
OD3D C5 F6    628 CMP ZCTMP ;YES-DO LOOP
OD3F F0 F3    629 BEQ COUNT1 ;STORE NEW VALUE OF POLARITY
OD41 85 F6    630 STA ZCTMP ;NO-ADD 1 TO CROSSINGS.
OD43 E6 02    631 INC CROSS ;UNCONDIT. JMP TO COUNT1
OD45 D0 ED    632 BNE COUNT1 ;RECOVER Y-REG
OD47 6B      633 COUNT3   PLA          ;FROM STACK
OD48 A8      634 TAY          ;LOAD ACCUM. WITH # OF O-CROSS
OD49 A5 02    635 LDA CROSS ;RETURN TO CALLER
IN65
OD4B 60      636 RTS       ;TURN OFF INTERRUPTS
OD4C          637 ;-----
OD4C 78      638 RESIRO   SEI          ;TURN OFF
OD4D AC F1 02 639 LDY SLOT    ;UART IRQ'S
OD50 A9 04    640 LDA #$04
OD52 99 8C C0  641 STA COM,Y
OD55 60      642 RTS

```

32

```

OD56      643 ; NOTE: IT IS VERY IMPORTANT THAT THE
OD56      644 ; FOLLOWING ROUTINE BE LOCATED IN
OD56      645 ; A FULL PAGE ($FF BYTES NO BOUNDARY
OD56      646 ; CROSSING) BECAUSE OF TIMING REQUIREMENTS
OD00      647     ORG $FF00:$100+* ; SKIP TO NEXT EVEN PAGE BOUNDARY
Y
OE00      648 ; -----
OE00      649 ; *** TOUCH TONE ROUTINE ***
OE00      650 ; MUST ALL BE IN ONE PAGE OF MEMORY
OE00      651 ; -----
OE00      652 ; 16-BYTE SINE LOOK-UP TABLE
OE00 40 59 653 SINE   BYT $40,$59
OE02 6C 7A 654 BYT $6C,$7A
OE04 7F 7A 655 BYT $7F,$7A
OE06 6C 5B 656 BYT $6C,$5B
OE08 40 28 657 BYT $40,$2B
OE0A 13 05 658 BYT $13,$05
OE0C 00 05 659 BYT $00,$05
OE0E 13 28 660 BYT $13,$2B
OE10      661 ; -----
OE10      662 ; TOUCH-TONE CONSTANTS.
OE10      663 ; (NEED TO BE ALTERED IF
OE10      664 ; CPU CLOCK ISN'T 1.023 MHZ)
OE10      665 ; RE-CALCULATED FOR 93 MACHINE CYCLES
OE10      666 ; 11/11/80
OE10      667 ; -----
OE10 5E 01 668 TABL   BYT $5E,$01
OE12 F1 01 669 BYT $F1,$01
OE14 04 01 670 D1    BYT $04,$01
OE16 C2 01 671 BYT $C2,$01
OE18 04 01 672 D2    BYT $04,$01
OE1A F1 01 673 BYT $F1,$01
OE1C 04 01 674 D3    BYT $04,$01
OE1E 26 02 675 BYT $26,$02
OE20 1F 01 676 D4    BYT $1F,$01
OE22 C2 01 677 BYT $C2,$01
OE24 1F 01 678 D5    BYT $1F,$01
OE26 F1 01 679 BYT $F1,$01
OE28 1F 01 680 D6    BYT $1F,$01
OE2A 26 02 681 BYT $26,$02
OE2C 3D 01 682 D7    BYT $3D,$01
OE2E C2 01 683 BYT $C2,$01
OE30 3D 01 684 DB    BYT $3D,$01
OE32 F1 01 685 BYT $F1,$01
OE34 3D 01 686 D9    BYT $3D,$01
OE36 26 02 687 BYT $26,$02
OE38 5E 01 688 D10   BYT $5E,$01
OE3A C2 01 689 BYT $C2,$01 ;#
OE3C 5E 01 690 D11   BYT $5E,$01
OE3E 26 02 691 BYT $26,$02 ;*
OE40      692 ; -----
OE40      693 ; TOUCHTONE GEN.
OE40      694 ;
OE40 86 1C 695 TONE   STX TEMPX ;SAVE X REG
OE42 84 1D 696 STY TEMPY ;SAVE Y REG
OE44 A2 04 697 LDX #4
OE46 86 EA 698 STX DLYH ;SET-UP INTERDIGIT DELAY

```

```

OE48 0A 699 MULT4  ASL          ;MULTIPLY
OE49 0A 700             ;DIGIT BY 4
OE4A AB 701             ;PUT OFFSET IN Y
OE4B B9 10 0E 702         LDA TABL,Y ;GET 1ST FRACTION
OE4E 85 E5 703             STA F1E
OE50 CB 704             INY
OE51 B9 10 0E 705         LDA TABL,Y ;GET 2ND INTEGER
OE54 85 E6 706             STA F1I
OE56 CB 707             INY
OE57 B9 10 0E 708         LDA TABL,Y ;GET 2ND FRACTION
OE5A 85 E7 709             STA F2F
OE5C CB 710             INY
OE5D B9 10 0E 711         LDA TABL,Y ;GET 2ND INTEGER
OE60 85 EB 712             STA F2I
OE62 EA 713             NEXT
OE63 EA 714             NOP
OE64 EA 715             NOP
OE65 EA 716             NOP
OE66 A5 E5 717             NEXT1
OE68 1B 718             CLC
OE69 65 E1 719             ADC FRAC1
OE6B 85 E1 720             STA FRAC1
OE6D A5 E6 721             LDA F1I
OE6F 65 E0 722             ADC INT1
OE71 85 E0 723             STA INT1
OE73 29 0F 724             AND #$F
OE75 AA 725             TAX
OE76 BD 00 0E 726             LDA SINE,X
OE79 4B 727             PHA
OE7A A5 E7 728             LDA F2F
OE7C 1B 729             CLC
OE7D 55 E4 730             ADC FRAC2
OE7F 85 E4 731             STA FRAC2
OE81 A5 E8 732             LDA F2I
OE83 65 E3 733             ADC INT2
OE85 85 E3 734             STA INT2
OE87 29 0F 735             AND #$F
OE89 AA 736             TAX
OE8A 6B 737             PLA
OE8B 1B 738             CLC
OEBC 7D 00 0E 739             ADC SINE,X
OEBF 29 F0 740             AND #$FO
OE91 09 04 741             ORA #$04
OE93 AC F1 02 742             LDY SLOT ;GET SLOT INDEX
OE96 99 B0 C0 743             STA DACBYT,Y ;STORE TO DAC
OE99 C0 E9 744             DEC DLYL
OE9B D0 C5 745             BNE NEXT
OE9D C0 EA 746             DEC DLYH
OE9F D0 C5 747             BNE NEXT1
OEA1 A9 74 748             LDA #$74
OEAD AC F1 02 749             LDY SLOT ;GET SLOT INDEX
OEAE 99 B0 C0 750             STA DACBYT,Y ;STORE TO DAC
OEAF A9 10 751             LDA #$10
OEAB 85 EA 752             DELAY2
OEAD C6 E9 753             DLY1
OEAF D0 FC 754             BNE DLY1 ;ROUTINE

```

LISA 2.5

CATPACK



```

OE81 C6 EA    755      DEC DLYH
OE83 D0 FB    756      BNE DLY1
OE85 A6 1C    757      LDX TEMPX      ;RESTORE X
OE87 A4 1D    758      LDY TEMPY      ;AND Y REGS
OE89 60       759      RTS

OEBA          760      -----
OEBA          761      ;THIS SECTION GETS A CHAR FROM THE KEYBD
OEBA          762      ;AND PRINTS BOTTOM LINES IF NECESSARY.
OEBA AD 00 CO 763      RDKEY   LDA KEYDAT      ;READ KEYBOARD
OEBD 30 02    764      BMI RDKEY1     ;YES, THEN GO TO RDKEY1
OEBF 10       765      CLC               ;NO-SEND ROUTINE BACK
OECC 60       766      RTS               ;WITH CARRY CLEAR TO MEAN NO C
HARACTOR
OECD 1 AD 10 CO 767      RDKEY1   LDA KEYCLR     ;CLEAR KEYBOARD STROBE AND STR
IP MSB
OECA 4 AD 00 CO 768      LDA KEYDAT      ;GET CHARACTER FROM KEYBOARD
OECD 38       769      SEC               ;AND RETURN TO ROUTINE
OECC 50       770      RTS               ;WITH CARRY SET TO MEAN GOT CH
AR.
OECD 9          771      ;-----

OECD 9 68      772      COMMD1   PLA           ;GET RETURN ADDRESS SO WE
OECA 85 95      773      STA CMDTAB     ;KNOW WHERE DATA STARTS
OECC 60          774      PLA           ;THEN STORE LOCATION IN
OECD 85 95      775      STA CMDTAB+1  ;THIS PAGE ZERO LOCATION
OECD 1 AD 10 CO 776      LDA KEYCLR     ;CLEAR KEYBOARD STROBE
OECD 2 20 BA OE 777      MENU2   JSR RDKEY     ;READ KEYBOARD
OEDE 5 90 FB    778      BCC MENU2      ;IF CARRY CLEAR, DO IT AGAIN
OEDE 7 85 5D    779      STA CMD        ;STORE IN ITS OWN LOCATION
OEAD 90 01       780      LDY #1         ;PUT 1 INTO Y
OEDE 81 95       781      CMD1   LDA (CMDTAB),Y  ;GET CHAR INDEXED BY Y
OEDE 9 3B        782      CMP #'?'      ;IS IT THE END OF COMMANDS
OEDE F0 21       783      BEQ CMD4      ;YES-THEN END
OEEL 1 C5 5D    784      CMP CMD        ;IS IT A COMMAND
OEES 3 F0 0C    785      BEQ CMD2      ;YES-GO OFF AND DO IT
OEES 5 20 0C 0F  786      JSR UPDAT1    ;INCREMENT
OEEB 20 0C 0F    787      JSR UPDAT1    ;CMDTAB LOCATION
OEEB 20 0C 0F    788      JSR UPDAT1    ;THREE TIMES
OEEE 4C DB 0E    789      JMP CMD1      ;INCREMENT Y ONCE MORE
OEFD 1 20 0C 0F  790      CMD2   JSR UPDAT1
OEFD 4 B1 95      791      LDA (CMDTAB),Y
OEFD 6 85 95      792      STA CMDOLOC
OEFB 20 0C 0F    793      JSR UPDAT1
OEFB 81 95       794      LDA (CMDTAB),Y
OEFD 85 98       795      STA CMDOLOC+1
OEFF 6C 97 00    796      JMP (CMDOLOC)
OF02 20 0C 0F    797      CMD4   JSR UPDAT1
OF05 A5 96       798      LDA CMDTAB+1
OF07 4B          799      PHA
OF08 A5 95       800      LDA CMDTAB
OF0A 4B          801      PHA
OF0B 60          802      RTS
OF0C E6 95       803      UPDAT1  INC CMDTAB
OF0E D0 02       804      BNE UPDAT2
OF10 E6 96       805      INC CMDTAB+1
OF12 60          806      UPDAT2  RTS
OF13             807      END

```

\*\*\*\*\* END OF ASSEMBLY

**Novation**

